

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-161587

(43)公開日 平成11年(1999)6月18日

(51)Int.Cl. <sup>8</sup>	識別記号	F I	
G 0 6 F 13/00	3 5 5	G 0 6 F 13/00	3 5 5
	3 5 4		3 5 4 D
3/00	6 5 1	3/00	6 5 1 E
	6 5 2		6 5 2 A
H 0 4 L 12/28		H 0 4 L 11/00	3 1 0 Z

審査請求 有 請求項の数 4 O L (全 10 頁)

(21)出願番号 特願平9-323403

(22)出願日 平成9年(1997)11月25日

(71)出願人 000211329

中国日本電気ソフトウェア株式会社  
広島県広島市南区稲荷町4番1号

(72)発明者 藤岡 博

広島県広島市南区稲荷町4番1号 中国日本電気ソフトウェア株式会社内

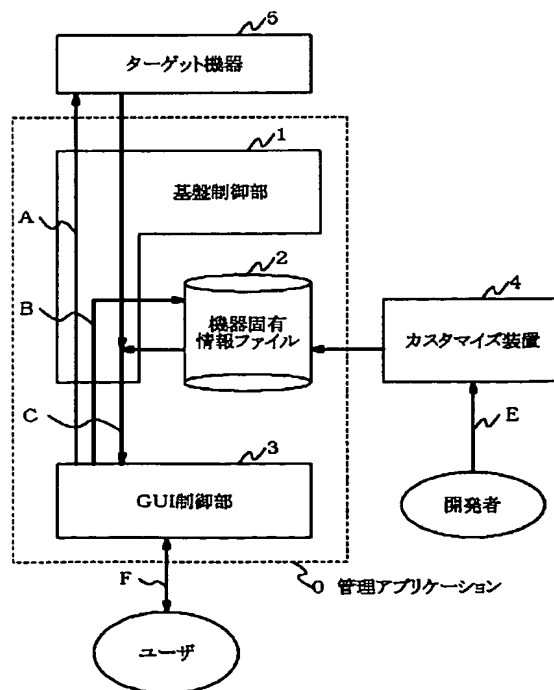
(74)代理人 弁理士 京本 直樹 (外2名)

(54)【発明の名称】 ネットワーク接続機器管理アプリケーション開発方式

(57) 【要約】

【課題】 ネットワーク接続機器（ハブ、ルータ等）の管理アプリケーションを開発を効率的にする。

【解決手段】 カスタマイズ装置4は、ターゲット機器5固有のカスタマイズデータを定義し機器固有情報ファイル2を作成する。基盤制御部1は、該機器固有情報ファイルを解析しターゲット機器固有のGUI構成データを取得する機器固有データ解析部とターゲット機器の最新のデータを取得するインタフェース自動発見部とを有する。GUI制御部3は、ターゲット機器固有のGUI構成データと取得したターゲット機器の最新データからGUIを作成し表示する。



**【特許請求の範囲】**

**【請求項1】** ネットワーク接続機器を管理するための管理アプリケーションを開発する場合のネットワーク接続機器管理アプリケーション開発方式において、前記ネットワーク接続機器固有のカスタマイズデータを定義し機器固有情報ファイルを作成するカスタマイズ装置と、該機器固有情報ファイルを解析し前記ネットワーク接続機器固有のG U I 構成データを取得する機器固有データ解析部と前記ネットワーク接続機器の最新のデータを取得するインタフェース自動発見部とを有する基盤制御部と、前記ネットワーク接続機器固有のG U I 構成データと取得した前記ネットワーク接続機器の最新データからG U I を作成し表示するG U I 制御部から構成されることを特徴とするネットワーク接続機器管理アプリケーション開発方式。

**【請求項2】** 前記カスタマイズデータは、前記ネットワーク接続機器に依存したG U I 上のメニュー定義とダイアログに表示する項目、及び前記ネットワーク接続機器のネットワークとのインタフェース、L E D の構成状態を含んで構成されることを特徴とする請求項1記載のネットワーク接続機器管理アプリケーション開発方式。

**【請求項3】** 前記インタフェース自動発見部は、定期的に前記ネットワーク接続機器の構成や状態をポーリングすることにより、前記ネットワーク接続機器のL E D の構成状態及び該L E D の色に変更があった場合に、該変化を前記G U I に反映させるように前記G U I 制御部を介して制御することを特徴とする請求項1記載のネットワーク接続機器管理アプリケーション開発方式。

**【請求項4】** 前記インタフェース自動発見部は、定期的に前記ネットワーク接続機器の構成や状態をポーリングすることにより、前記ネットワーク接続機器のネットワークとのインタフェースの構成状態に変更があった場合に、該変化を前記G U I に反映させるように前記G U I 制御部を介して制御することを特徴とする請求項1記載のネットワーク接続機器管理アプリケーション開発方式。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** 本発明は、管理アプリケーション開発方式に関し、特に多くのターゲット機器の管理アプリケーションの開発を短期間で効率よく実現できる開発方式に関する。

**【0002】**

**【従来の技術】** ルータやハブといったネットワーク接続機器を管理するための管理アプリケーションを開発し、各機器のフロントパネル上のポートやL E D 等の実装状態をそっくりそのままグラフィック化し、各機器に対する操作をグラフィック画面上から行うG U I を提供することは、リモートに存在する機器があたかも目の前に存

在するかのごとく表現でき、機器の管理という面で直感的でわかりやすく、またインパクトのある方法である。

**【0003】** 従来、この種の管理アプリケーションの開発は、図9に示すように個々のターゲット機器に特化した開発方法であった。

**【0004】**

**【発明が解決しようとする課題】** 上述した従来のネットワーク接続機器管理アプリケーション開発方式では、個々のターゲット機器に特化した管理アプリケーション開発方式のため、多くのベンダから様々な機種が出荷され、しかもそれらのライフサイクルが短い現状では、開発効率及び保守効率が悪いという欠点があった。

**【0005】** 本発明の目的は、共通の基盤とG U I を持ち、この基盤を利用することにより、新たな管理アプリケーションを開発する際にも、一から開発するのではなく、機器固有情報ファイルを作成するだけで開発できるコンパクトな管理アプリケーション開発方式を提供することにある。

**【0006】**

**【課題を解決するための手段】** 本願の第1の発明は、ネットワーク接続機器を管理するための管理アプリケーションを開発する場合のネットワーク接続機器管理アプリケーション開発方式において、前記ネットワーク接続機器固有のカスタマイズデータを定義し機器固有情報ファイルを作成するカスタマイズ装置と、該機器固有情報ファイルを解析し前記ネットワーク接続機器固有のG U I 構成データを取得する機器固有データ解析部と前記ネットワーク接続機器の最新のデータを取得するインタフェース自動発見部とを有する基盤制御部と、前記ネットワーク接続機器固有のG U I 構成データと取得した前記ネットワーク接続機器の最新データからG U I を作成し表示するG U I 制御部から構成されることを特徴とする。

**【0007】** 本願の第2の発明は、第1の発明における前記カスタマイズデータは、前記ネットワーク接続機器に依存したG U I 上のメニュー定義とダイアログに表示する項目、及び前記ネットワーク接続機器のネットワークとのインタフェース、L E D の構成状態を含んで構成されることを特徴とする。

**【0008】** 本願の第3の発明は、第1の発明における前記インタフェース自動発見部は、定期的に前記ネットワーク接続機器の構成や状態をポーリングすることにより、前記ネットワーク接続機器のL E D の構成状態及び該L E D の色に変更があった場合に、該変化を前記G U I に反映させるように前記G U I 制御部を介して制御することを特徴とする。

**【0009】** 本願の第4の発明は、第1の発明における前記インタフェース自動発見部は、定期的に前記ネットワーク接続機器の構成や状態をポーリングすることにより、前記ネットワーク接続機器のネットワークとのインタフェースの構成状態に変更があった場合に、該変化を

前記GUIに反映させるように前記GUI制御部を介して制御することを特徴とする。

【0010】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して詳細に説明する。

【0011】図1は、本発明の一実施の形態を示すブロック図である。

【0012】図1において、基盤制御部1は、

- ・機器固有情報ファイル2から解析したGUI構成データから、必要とするターゲット機器取得データを取得し、GUI制御部3へ送ったり（図1-C）、
- ・GUI制御部3から設定要求（具体的には、ターゲット機器のリセットなどGUI画面からの設定処理があった場合）があると（図1-F）、設定要求からターゲット機器設定データを作成して、ターゲット機器5に設定したり（図1-A）、
- ・GUI保存データを機器固有情報ファイル2へ保存する（図1-B）機能を持つ。

【0013】機器固有情報ファイル2は、管理アプリケーション0を開発する際に、管理アプリケーション開発者が、ターゲット機器毎のカスタマイズデータをカスタマイズ装置4で定義して、生成されたファイルである。また、GUI画面上で編集されたGUI保存データを保存する情報ファイルである。

【0014】GUI制御部3は、基盤制御部1が機器固有情報ファイル2から解析したGUI構成データと、基盤制御部1がターゲット機器5から取得したターゲット機器取得データからGUIを作成し、表示する。

【0015】カスタマイズ装置4は、開発者が管理アプリケーションを開発する際に、ターゲット機器の固有のカスタマイズデータの定義をGUI上で容易に行える機能を持つ。

【0016】本装置を利用することにより、容易に新しいターゲット機器に対応できる。

【0017】図2は、本発明の開発環境イメージ図である。開発環境の観点から着目すると、基盤制御部とGUI制御部を共通化することにより、ターゲット機器毎の分析を行い、機器固有情報ファイルを作成するだけで、管理アプリケーションを作成できる特徴を持つ。

【0018】図3は、本発明の基盤制御部1の構成図である。

【0019】管理プロトコル通信部31は、ターゲット機器と実際に適切なプロトコルを用いて通信する部分である。例えば、SNMPプロトコルを管理プロトコルとして使用する場合には、SNMPデータのエンコード／デコード処理を行いデータ設定／取得を行う。データを設定する場合、ターゲット機器設定データを作成し、データを取得する場合、ターゲット機器取得データを作成する。

【0020】インタフェース自動発見部32は、ターゲ

ット機器のインタフェースとそのインタフェースに関連するLEDの構成を解析する部分である。機器固有情報ファイル中のインタフェースに関する構成情報を機器固有データ解析部33で解析したGUI構成データと管理プロトコル通信部31から取得したターゲット機器取得データから、どのような種別のインタフェース（LED）が、どのような位置にあるかの構成を発見するものである。

【0021】機器固有データ解析部33は、カスタマイズ装置より生成された機器固有情報ファイルと汎用データベース34のデータ（以下、GUI汎用データと云う。）を解析して、GUI構成データを作成する。インタフェース自動発見部とGUI-IF部35に要求された必要なGUI構成データを送る。また、GUI保存データに関する機器固有情報ファイルの書き込み／読み込み処理を行う。

【0022】汎用データベース34は、GUI汎用データが格納されている部分である。カスタマイズ装置でカスタマイズデータを設定する際に、あらかじめ設定されているデータである。

【0023】GUI-IF（インタフェース）部35は、基盤制御部1とGUI制御部3とのインタフェース部である。管理アプリケーション実行時において、GUI作成時には、機器固有データ解析部33からのGUI構成データをGUI制御部3に送る。また、GUI制御部から設定要求があった場合、設定内容を管理プロトコル通信部31に送ったり、GUI制御部3から取得要求があった場合、管理プロトコル通信部31からターゲット機器取得データをGUI制御部3へ送る機能をもつ。

【0024】・GUI汎用データ汎用性のあるGUIメニューやダイアログの定義（例えば、ターゲット機器が標準でサポートしているシステム情報やインタフェース情報など）や、汎用性のあるインタフェースやLEDビットマップファイルなどが格納されている。

【0025】次に本発明の実施の形態の動作について説明する。

【0026】図4は、カスタマイズ装置を用いた機器固有情報ファイルの作成についてのフローチャートである。ここでは、管理アプリケーション開発者がカスタマイズデータの定義を行う。

【0027】・（ステップ41）ターゲット機器を識別するためのベンダ識別子を定義する。

【0028】・（ステップ42）ターゲット機器のパネルをイメージしたビットマップファイルを作成し、定義する。ここで作成したビットマップファイルは、管理アプリケーションのメインウィンドウであるパネルウィンドウ画面として使用される。作成したビットマップファイル名をカスタマイズ装置で定義する。

【0029】・（ステップ43）インタフェースの情報を定義する。インタフェースの構成を識別するための情

報である。具体的には、インタフェースのタイプや、パネルウィンドウ上の位置や、インタフェース形状をイメージしたビットマップファイル名などである。

【0030】・(ステップ44) パネルウィンドウのメニューの定義を行う。どのようなメニューの構成にするか考えて、メニューのラベルとニーモニックを定義する。

【0031】・(ステップ45) 表示情報の設定とメニューとのリンクを行う。表示情報の設定は、表示するダイアログにどの項目をどのようなレイアウトで表示するかを指定する。表示項目には、ターゲット機器のサポートしている情報を指定する。具体的には、SNMPプロトコルを使用している場合は、MIB (Management Information Baseの略。SNMPプロトコルで使用するツリー構造のデータ定義のこと) 名となり、ダイアログ単位で表示情報の定義を行い、ラベル名と表示項目名を指定する。その後、どのメニューからどのダイアログが表示されるかの指定を行う。

【0032】・(ステップ46) インタフェースのリンク状態やコリジョン状態などを示すLEDの定義である。LEDの表示用ビットマップファイルとパネルウィンドウに表示する位置の定義を行う。

【0033】・(ステップ47) ポーリング情報を定義する。管理アプリケーションのインタフェース自動発見部は指定されたポーリング情報について定期的にポーリング処理を行う。例えば、LED情報を定義する場合、LED状態の判断対象となる情報と、その情報の値と表示するLEDの色の対応付けの設定を行う。ここで設定した情報から、パネルウィンドウにLEDが表示されて、ポーリング時にLED状態をターゲット機器より取得して、定義した情報から判断した色を表示する。また、ターゲット機器が、拡張ボードの搭載や、スタック構成可能で、管理アプリケーション起動中に、機器や機器のインタフェースの構成に変更があった場合など、パネルウィンドウも変更する必要がある際に定義する。

【0034】以上のような処理フローで、管理アプリケーションを開発する際には、カスタマイズ装置を使用して、機器固有情報ファイルを作成することにより、容易に新しいターゲット機器に対応できる。

【0035】図5は、管理アプリケーション起動時および動作中のフローチャートである。フローチャートの説明と、各ステップが実行環境のどの部分に働くか、図1、図3を用いて説明する。

【0036】・(ステップ51) ターゲット機器5より、基盤制御部1の管理プロトコル通信部31を介して、機種を判別するためにベンダ識別子を取得する。

【0037】・(ステップ52) 取得したベンダ識別子を用いて、基盤制御部の機種固有データ解析部33は、機種を特定し、該当する機器固有情報ファイル2を

読み込む。

【0038】・(ステップ53) 読み込んだ機器固有情報ファイルを用いて、基盤制御部のGUI-iff部35は、パネルイメージのビットマップファイルを読み込む。

【0039】・(ステップ54) 基盤制御部のインタフェース自動発見部32は、機器固有情報ファイル中のインタフェースに関する構成情報を機器固有データ解析部33で解析したGUI構成データと管理プロトコル通信部31から取得したターゲット機器取得データから、どのような種類のインタフェースが、どういう位置にあるかの構成を発見する。

【0040】・(ステップ55) 機器固有データ解析部により機器固有情報ファイルから前回終了時に書き込まれたGUI保存データとインタフェース自動発見部から解析された今回起動時のインタフェースの構成が同じかどうかを調べる分岐である。初回の管理アプリケーション起動時の場合(GUI保存データがない)、ステップ57の処理へ進む。

【0041】・(ステップ56) 機器固有データ解析部33は、機器固有情報ファイル中のGUI保存データ中のインタフェース情報(インタフェースタイプやパネルウィンドウ上の座標など)を読み込む。

【0042】・(ステップ57) 機器固有データ解析部は、インタフェース自動発見処理(ステップ54)のデータをもとにGUIに表示するためのインタフェース情報をGUI-iff部35へ設定する。

【0043】・(ステップ58) 機器固有データ解析部により、機器固有情報ファイルのGUI保存データをもとにGUIに表示するためのインタフェース情報をGUI-iff部35へ設定する。前回、管理アプリケーション起動時と同様のインタフェース構成である。

【0044】・(ステップ59) ステップ55と同様に、機器固有データ解析部33により機器固有情報ファイルから前回終了時に書き込まれたGUI保存データとインタフェース自動発見部32から解析された今回起動時のインタフェースの構成中のLED構成が同じかどうかを調べる分岐である。初回の管理アプリケーション起動時の場合(GUI保存データがない)、ステップ61の処理へ進む。

【0045】・(ステップ60) 機器固有データ解析部33は、機器固有情報ファイル中のGUI保存データ中のLED情報(LEDのビットマップファイル名やパネル画面上の座標など)を読み込む。

【0046】・(ステップ61) 機器固有データ解析部33は、インタフェース自動発見処理(ステップ54)のデータをもとにGUIに表示するためのLED情報をGUI-iff部35へ設定する。

【0047】・(ステップ62) 機器固有データ解析部33は、機器固有情報ファイルのGUI保存データを

もとにGUIに表示するためのLED情報をGUI-ifu部35へ設定する。前回、管理アプリケーション起動時と同様のLED構成である。

【0048】・(ステップ63) GUI-ifu部35からGUI構成データを受け取ったGUI制御部3は、インタフェース情報、LED情報、パネルイメージビットマップをパネルウィンドウに設定し、ウィンドウ、メニューなどのGUIを形成し、表示する。

【0049】・(ステップ64) GUI制御部3には、GUI-ifu部35から受け取ったカスタマイズ装置で定義したポーリング情報をターゲット機器より取得する。前回取得したデータを退避しておく。

【0050】・(ステップ65) GUI制御部3により、前回、ターゲット機器より取得したポーリング情報のデータと今回取得したデータとで、変更があるかどうか調べる分岐である。前回と同じ(Yes)場合、再びポーリング処理(ステップ64)を行い、前回と異なる(No)場合、パネルウィンドウ表示(ステップ63)の処理を行い、変更のあったポーリング情報のデータをもとにパネルウィンドウを表示し直す。例えば、ポーリング情報にLEDやインタフェースの状態を定義した場合、定義した色でビットマップファイルを表示し直す。また、機器構成やインタフェース構成をポーリング情報に設定している場合、実機の構成と同様のイメージをパネルウィンドウに表示し直す。

【0051】・(ステップ66) GUI制御部3は、管理アプリケーション終了時に、インタフェースやLED構成などのデータを、機器固有情報ファイルにGUI保存データとして書き込む。

【0052】以上のような処理フローで、機器固有情報ファイルのみの作成で管理アプリケーションが動作できる。

【0053】

【実施例】次に、本発明の実施例の構成について図面を参照して説明する。

【0054】図6は、本発明の一実施例を示す動作説明図である。

【0055】ターゲット機器71は、今から管理アプリケーションを開発しようとしている機器である。ターゲット機器の特徴は、以下のようになっている。

【0056】・管理プロトコルとして、SNMPエージェントを搭載している。

【0057】・インタフェースの構成は、100BASE-Tインタフェースが24個、SERIALインタフェースが1個、ATMインタフェースが1個、FDDIインタフェースが1個で構成されている。

【0058】・拡張スロットを2スロット搭載しており、運用中、スロットにインタフェースボードを挿入することで構成を変更できる。

【0059】・インタフェースボードは、ATMボード

とFDDIボードの2種類ある。

【0060】・100BASE-Tインタフェースには、それぞれに対応したLEDがあり、通常はオレンジ色で、インタフェースがネットワーク混雑状態の場合に、緑色に点灯する。また、ターゲット機器が各インタフェースに一意に設定している番号(以下、インタフェース番号と云う)は、100BASE-T、シリアル、ATM、FDDIインタフェースに昇順に割り当てられているものとする。

【0061】ターゲット機器のカスタマイズデータ72を用いて、本発明のカスタマイズ装置73で機器固有情報ファイルを生成し、管理アプリケーション74を完成させる。

【0062】次に、本発明の実施例の動作について、カスタマイズ装置の処理と管理アプリケーションの動作処理について説明する。

(1) カスタマイズ装置の処理

図6、図7、図8とカスタマイズ装置より機種固情報ファイル作成の処理フロー(図4)を用いて詳細に説明する。

【0063】図7は、カスタマイズデータ(図6の72)からカスタマイズ装置を用いて生成した実施例に関連する機器固有ファイルの構造例である。

【0064】1. ターゲット機器のベンダ識別子を定義(図4、ステップ41)ターゲット機器を識別するためのベンダ識別子を定義する(図7-F)

2. パネルビットマップ定義(図4、ステップ42)ターゲット機器のパネルをイメージしたビットマップファイルを作成し、ファイル名を設定する。(図7-D)

3. インタフェース情報定義(図4、ステップ43)ターゲット機器がサポートしているMIBの定義から、インタフェース形状を識別できるMIB(タイプ識別MIB(図7-A、②))とインタフェース番号識別できるMIB(インタフェース番号識別MIB(図7-A、④))を特定する。

【0065】インタフェース位置が固定のインタフェースの位置を設定する(図7-A、①)。本実施例では、100BASE-TインタフェースとSERIALインタフェースが該当する。座標は、パネルイメージビットマップファイル上の座標であり、インタフェース番号順に列挙していく。

【0066】インタフェースのタイプを識別するため、タイプ(図7-A、③)を設定する。実際のインタフェースタイプ(10BASE-T等)とターゲット機器が持っているインタフェースタイプ識別MIBの値との関連付けをする。例えば、ATMインタフェースのインタフェース識別MIBの値が、“1”である場合には、“ATM:1”となる。インタフェースタイプは、カスタマイズ装置により、選択できる。

【0067】インタフェースの拡張用ボードの構成を設

定する。ターゲット機器がサポートしているMIBの定義から、ボードのタイプを識別できるMIB(図7-A, ⑧)とボードの番号を識別できるMIB(図7-A, ⑤)を特定する。

【0068】ポートタイプ(図7-A, ⑥)には、図7-A, ⑧の値とインタフェースのタイプとそのインタフェースのパネルイメージビットマップファイル上のボード位置を原点とした座標を順に列挙する。

【0069】次にボードの位置(図7-A, ⑦)を設定する。パネルイメージビットマップファイル上でのボードを表示する座標を設定する。

【0070】4. メニュー定義(図4, ステップ44)、表示情報定義とメニューとのリンク(図4, ステップ45)メニューの設定とダイアログの表示内容を設定する。メニュー名とニックネームとダイアログIDを設定する(図7-C, ①)。ダイアログIDは、どのダイアログを表示するかの識別子である。

【0071】ダイアログのタイトルと表示内容を設定する。ラベル名(項目内容)と表示するMIB名を設定する。例えば、図8-①のようなダイアログを表示する場合、機器固有情報ファイルの内容は、図8-②のようなデータをカスタマイズ装置が生成する。

【0072】5. LED情報定義(図4, ステップ46)LEDのパネルイメージビットマップファイル上の座標をLEDの番号(インデックス)順に列挙する(図7-B, ①)。

【0073】LEDビットマップファイル名を設定する(図7-B, ②)。

【0074】ターゲット機器がサポートしているMIBの定義から、LEDの番号を識別できるMIB(図7-B, ③)を特定する。ポーリング情報でLEDの状態を取得する場合、どのLEDかを特定するための値である。

【0075】6. ポーリング情報定義(図4, ステップ47)ポーリングする対象のMIBを指定する。実施例では、インタフェースとLEDの状態をポーリングする設定について説明する。

【0076】インタフェース状態を設定する例では、図7-E, ①にインタフェース状態を識別するMIBを設定する。例えば、インタフェースのUP/DOWN状態(物理的に特定のインタフェースが接続可能な状態(UP)であるか、不可能な状態(DOWN)であるかを表す。)などがある。次に、図7-E, ②に、インタフェース状態識別のMIBの値と状態色を対応付ける。例えば、インタフェースのUP/DOWN状態のポーリングでは、UP状態のインタフェースの色を黒に、DOWN状態のインタフェースの色を赤に設定すると、管理アプリケーション動作中に実際のインタフェースが何らかの原因でDOWN状態になった場合、ポーリング時に検出してそのインタフェースビットマップを赤色に表示し、

GUI画面上でインタフェースの状態を視覚的に検出することが可能となる。

【0077】LED状態の場合もインタフェース状態と同様である。実施例では、LEDがあるインタフェースのネットワーク混雑状態を識別するMIB(図7-E, ③)を設定する。インタフェースが正常な状態のときのMIBの値とLEDの色をオレンジ色に、ネットワーク混雑状態のときのMIBの値とLEDの色を緑色に設定する(図7-E, ④)。設定すると管理アプリケーション動作中に、ネットワーク混雑状態のインタフェースをGUI画面上で視覚的に検出することが可能となる。

#### (2) 管理アプリケーションの動作処理

次に、図7のカスタマイズ装置で生成した機種固有情報ファイルが管理アプリケーションの動作処理(図5)にどう関連するか詳細に説明する。

【0078】1. 機器固有情報ファイルの読み込み(図5, ステップ52)取得したベンダ識別子より、基盤制御部の機種固有データ解析部33は、機器固有情報ファイル中のベンダ識別子情報(図7-F)を読み込み、一致したものをカレントとみなし、その機器固有情報ファイルを読み込む。

【0079】2. パネルイメージビットマップの読み込み(図5, ステップ53)読み込んだ機器固有情報ファイルを用いて、基盤制御部のGUI-iff部35は、機器固有情報ファイル中のパネルイメージビットマップファイル(図7-D)を取得し、指定のビットマップファイルを読み込む。

【0080】3. インタフェース自動発見処理(図5, ステップ54)GUI-iff部35は、機器固有データ解析部33が読み込んだ機器固有情報ファイルと管理プロトコル通信部31より取得したターゲット機器取得データより、インタフェースの自動発見処理を行う。以下に処理フローを簡条書きに示す。

【0081】・タイプ識別MIB(図7-A, ②)をターゲット機器より取得する。取得したデータは、現在、ターゲット機器に実装しているインタフェース数だけあり、インタフェースの形状を識別できる情報を持ったリストである。(インタフェース番号順に昇順になっている。)これによって、インタフェースの数が判断できる。

【0082】・インタフェース番号MIB(図7-A, ④)をターゲット機器より取得する。取得したデータは、タイプ識別MIBと順に対応している各インタフェース固有の番号のリストである。これによって、各インタフェースを一意に識別できる。

【0083】・タイプ(図7-A, ③)の情報を読み込む。タイプ識別MIBで取得した値は、各ベンダで定義しているものがある。ここでは、タイプ識別MIBで取得した値と読み込んだ上記の情報から各インタフェースの形状を判断する。これによって、各インタフェースの

形状が判断できる。

【0084】・位置(図7-A, ①)情報を読み込む。ここには、固定インタフェースの位置がインタフェース番号順に設定されているので、データがあるまで座標を決定していく。実施例の場合、この時点で100BASE-Tインタフェース24個分とシリアルインタフェース1個のタイプと位置が決定される。

【0085】・ボードタイプ識別MIB(図7-A, ⑧)をターゲット機器より取得する。取得したデータは、現在、ターゲット機器に実装しているボードの数だけあり、ボードのタイプを識別できる情報を持ったリストである。実施例の場合、ATMインタフェースとFDIインタフェースのボードのタイプが格納してある。これによって、ボードを実装している個数とタイプが判断できる。

【0086】・ボード番号識別MIB(図7-A, ⑤)をターゲット機器より取得する。取得したデータは、ボードタイプ識別MIBと順に対応している各ボード固有の番号のリストである。これによって、実装されているボードを一意に識別できる。

【0087】・ボード位置(図7-A, ⑦)情報を読み込む。ここには、指定のボード番号がパネルビットマップ上のどの位置にあるか設定されているので、取得したボード番号MIBの値とボードタイプ識別MIBの値から、どのボードがどの位置にあるか判断できる。

【0088】・ボードタイプ(図7-A, ⑥)情報を読み込む。ここでは、どのボードタイプにどのインタフェースが実装されて、ボードを原点としてどの位置にあるかが設定されている。既に、ボード位置とタイプがわかっているので、ボードに実装しているインタフェースの位置とタイプを判断できる。

【0089】・番号識別MIB(図7-B, ③)をターゲット機器より取得する。取得したデータは、現在、ターゲット機器に実装しているLEDの数だけあり、ターゲット機器に一意な昇順に並んだLED番号のリストである。これにより、LEDの数を判断できる。

【0090】・位置(図7-B, ①)情報を読み込む。ここでは、LEDの位置情報が昇順に列挙されており、各LEDのパネルビットマップ上の位置を判断できる。

【0091】・ビットマップ(図7-B, ②)の情報を読み込む。ここでは、表示するLEDのビットマップファイル名を判断できる。なお、インタフェースのビットマップに関しては、あらかじめ汎用データベース(図3, 34)に、GUI汎用データをして登録してあるので、それを使用する。

【0092】以上で、インタフェースの自動発見処理を行うことができる。

【0093】4. パネルウィンドウ表示(図5, ステップ63)ここでは、メニューとダイアログの作成及び動作について説明する。

【0094】・メニュー名(図7-C, ①)の情報を読み込む。ここでは、メニュー名、ニック、表示するダイアログを識別するダイアログIDを設定している。これにより、メニューを作成し、選択された場合、どのダイアログを表示するかを判断できる。

【0095】・ダイアログ(図7-C, ②)の情報を読み込む。ここでは、表示するダイアログの内容が設定されている。表示するラベル名とターゲット機器より取得するMIB名が設定されている。これにより、表示するダイアログの内容を判断できる。また、2. のパネルイメージビットマップの読み込みと3. の自動発見処理からGUI-iff部がGUI構成データを生成し、GUI制御部により、インタフェース情報、LED情報とパネルイメージビットマップをパネルウィンドウに設定し表示する。

【0096】5. ポーリング処理(図5, ステップ64)機種固有情報ファイルのポーリング情報(図7-E)より、情報を取得して、ポーリングを行う。

【0097】・インタフェース状態識別MIB(図7-E, ①)をターゲット機器より取得する。取得したデータは、現在、ターゲット機器に実装されているインタフェースの状態が格納されているインタフェース番号順に昇順になっているリストである。

【0098】・インタフェース状態色(図7-E, ②)の情報を読み込む。ここでは、インタフェース状態識別MIBで取得した値とパネルウィンドウに表示するインタフェースビットマップファイルの色とが対応付けられ、表示するインタフェースビットマップファイルの色を判断できる。

【0099】・LED状態識別MIB(図7-E, ③)や、LED状態色(図7-E, ④)もインタフェースの場合と同様である。

【0100】以上のことから、管理アプリケーションを開発する際に、本発明のカスタマイズ装置を使用することにより、ターゲット機器を分析した情報を容易に設定することができる。

【0101】また、カスタマイズ装置によって生成した機器固有情報ファイルを使用するだけで、複数のターゲット機器が存在する場合でも、共通化された基盤制御部とGUI機能で管理アプリケーションがコンパクトに作成でき、動作できる。

【0102】

【発明の効果】以上説明したように、本発明は、カスタマイズ装置を用いて、機器固有情報ファイルを作成するだけで新しいターゲット機器の管理アプリケーションを開発できるようにしたことにより、生産性が向上し、ライフサイクルが短いターゲット機器の管理アプリケーション開発に対して非常に有効になる効果があるとともに、全てのモジュールが共通化されているため、管理アプリケーションを保守する際に、保守効率がよくなる効

果もある。

【図面の簡単な説明】

【図 1】 本発明の一実施の形態を示すブロック図である。

【図 2】 本発明の開発環境イメージ図である。

【図 3】 本発明の基盤制御部 1 の構成図である。

【図 4】 本発明の機器固有情報ファイルの作成についてのフローチャートである。

【図 5】 本発明の管理アプリケーションの起動時および動作中のフローチャートである。

【図 6】 本発明の一実施例を示す動作説明図である。

【図 7】 本発明の機器固有ファイルの構造例を示す図である。

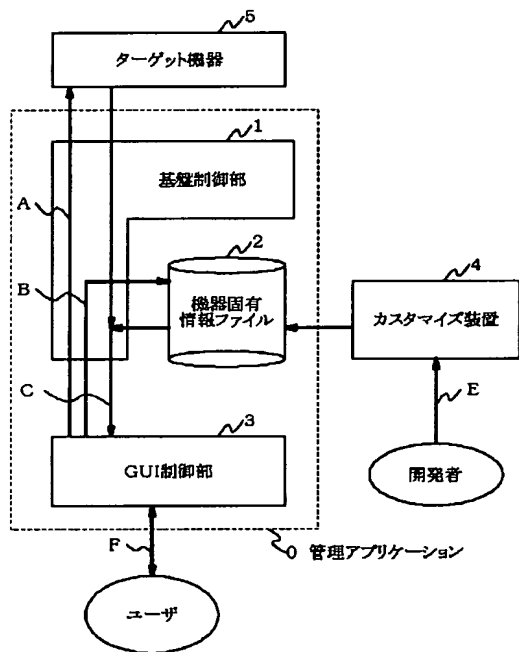
【図 8】 本発明のダイアログ表示例を示す図である。

【図 9】 従来のネットワーク接続機器管理アプリケーション開発方式を示す図である。

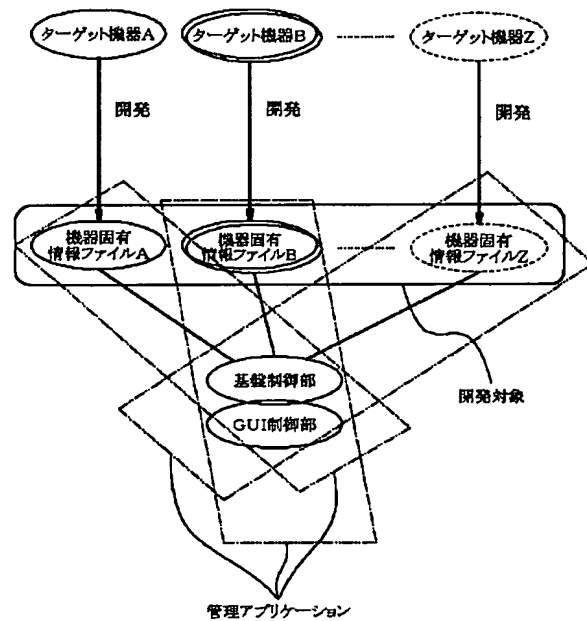
【符号の説明】

- 0, 7 4 管理アプリケーション
- 1 基盤制御部
- 2 機器固有情報ファイル
- 3 GUI 制御部
- 4, 7 3 カスタマイズ装置
- 5 ターゲット機器
- 3 1 管理プロトコル通信部
- 3 2 インタフェース自動発見部
- 3 3 機器固有データ解析部
- 3 4 汎用データベース
- 3 5 GUI - i f 部

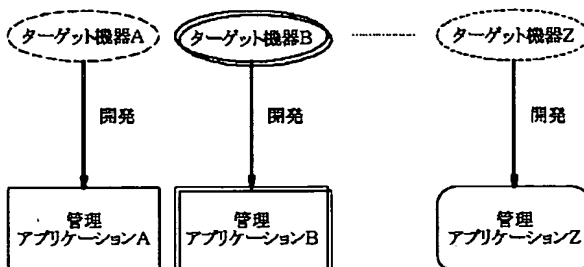
【図 1】



【図 2】

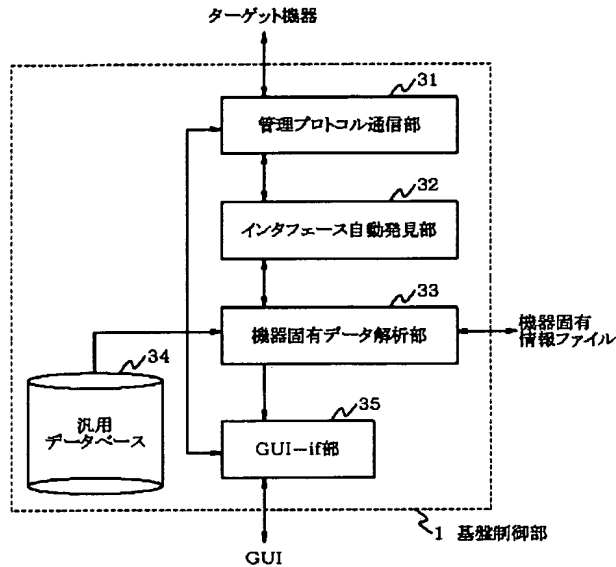


【図 9】

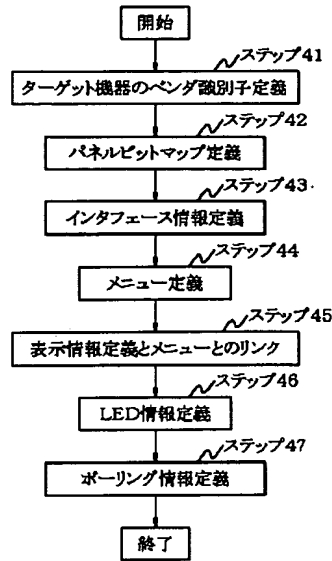




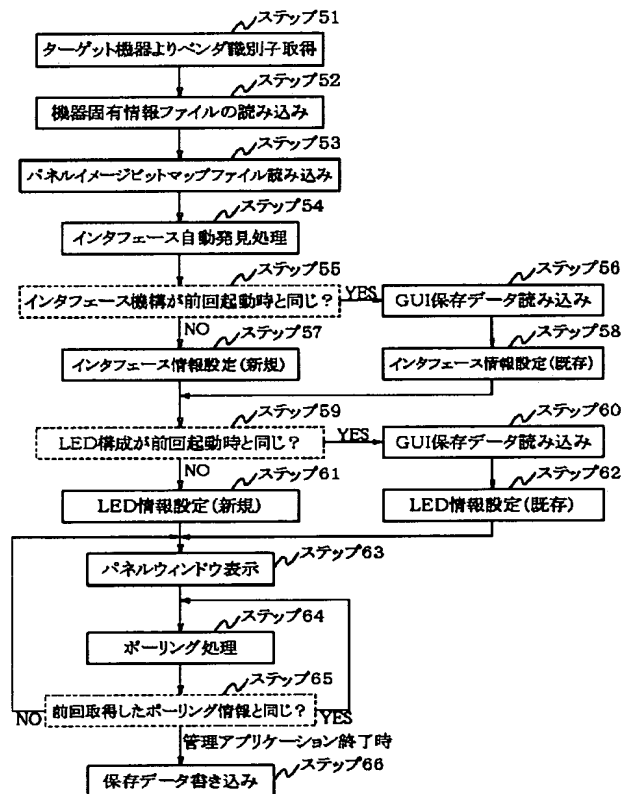
【図3】



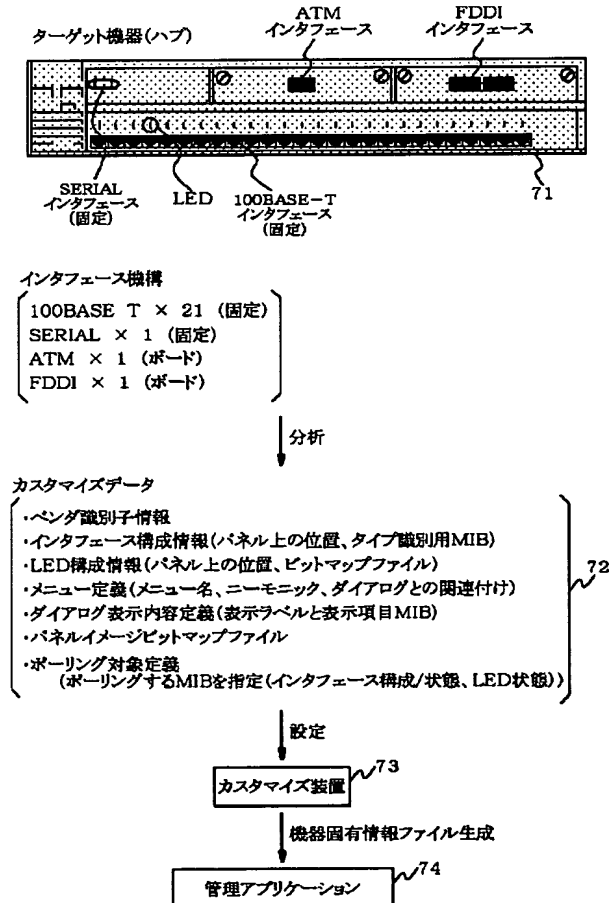
【図4】



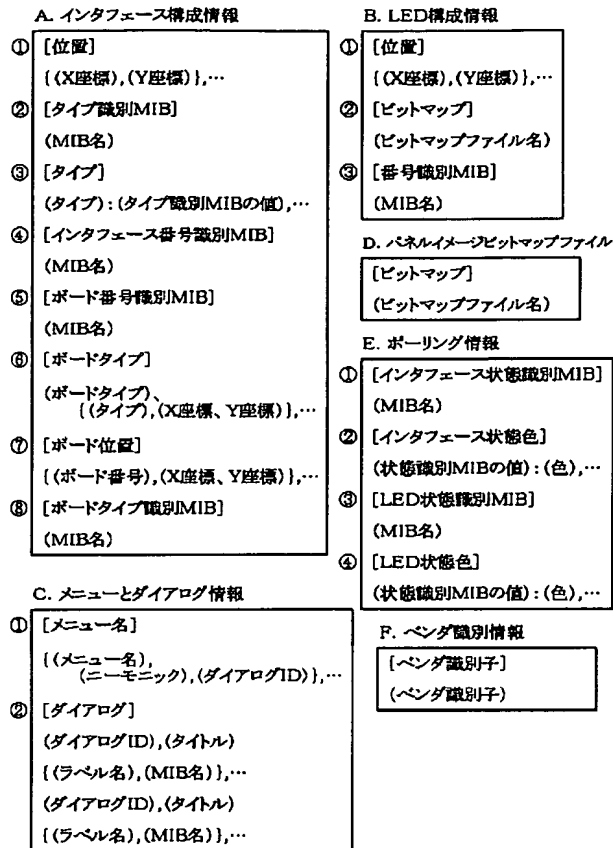
【図5】



【図6】



【図7】



【図8】

## ①メニューとダイアログ

システム(S)

システム情報

システム情報	XXX-HUB System Information
オブジェクトID	1.3.4.5.6.7.8.9
連絡先	Admin
システム名	XXX-HUB
設定場所	XXE M10F

OK

## ②機器固有情報ファイルの内容

[メニュー名]  
システム情報,S,1000  
[ダイアログ]  
1000;システム情報  
システム情報,(MIB名)  
オブジェクトID,(MIB名)  
連絡先,(MIB名)  
システム名,(MIB名)  
設定場所,(MIB名)